CLAIMS

1. A computer program product, tangibly embodied in an information carrier, comprising instructions operable to:

compile a procedural source code program to generate a compiled program having instructions to create a runtime data type having a compound structure of referenced data types and having instructions to use the runtime data type;

execute the instructions to create a runtime data type having a compound structure by creating a runtime data type definition from the compound structure of referenced data types by resolving the referenced data types bottom up into data types known at runtime;

create a data object having the runtime data type; and

perform type checking on uses of the data object at runtime according to the runtime data type.

2. The product of claim 1, wherein:

instructions operable to create a data object having the runtime data type comprise instructions operable to assign the runtime data type to a data object.

3. The product of claim 1, wherein:

the data types known at runtime comprise data types defined by type definitions generated by compiling the source code program.

4. The product of claim 1, wherein:

the instructions to create a runtime data type and the instructions to use the runtime data type are executed by a virtual machine; and

type definitions for the data types known at runtime are stored in a storage area managed by the virtual machine.

5. The product of claim 4, wherein:

the runtime type definition is stored in a local area for the computer program managed by the virtual machine.

6. The product of claim 1, wherein:

the runtime type definition is a type object.

7. The product of claim 6, wherein:

the type object is managed as an element in a class hierarchy.

8. The product of claim 6, wherein:

the type object is subject to garbage collection.

9. The product of claim 6, wherein:

the type object is referenced programmatically in the source code program by a handle and not by a name.

10. A computer system comprising:

a compile-time executable software module operable to compile a procedural source code program to generate a compiled program having instructions to create a runtime data type having a compound structure of referenced data types and having instructions to use the runtime data type; and

a runtime executable software module operable to:

execute the instructions to create a runtime data type having a compound structure by creating a runtime data type definition from the compound structure of referenced data types by resolving the referenced data types bottom up into data types known at runtime,

create a data object having the runtime data type, and

perform type checking on uses of the data object at runtime according to the runtime data type.

11. The system of claim 10, wherein:

creating a data object having the runtime data type comprises assigning the runtime data type to a data object.

12. The system of claim 10, wherein:

the data types known at runtime comprise data types defined by type definitions generated by compiling the source code program.

13. The system of claim 10, wherein:

runtime executable software module is a virtual machine.

14. The system of claim 10, wherein:

the new type object is a type object.

15. A method comprising:

compiling a procedural source code program to generate a compiled program having instructions to create a runtime data type having a compound structure of referenced data types and having instructions to use the runtime data type;

executing the instructions to create a runtime data type having a compound structure by creating a new data type definition from the compound structure of referenced data types by resolving the referenced data types bottom up into data types known at runtime;

creating a data object having the runtime data type; and

performing type checking on uses of the data object at runtime according to the runtime data type.

16. The method of claim 15 wherein

creating a data object having the runtime data type comprises assigning the runtime data type to a data object.

17. The method of claim 15, wherein:

the data types known at runtime comprise data types defined by type definitions generated by compiling the source code program.

18. The method of claim 15, wherein:

the new type object is a type object.